VAULT REFERENCE COPY

UCRL-85679 Viewgraphs
PREPRINT

THE INFLUENCE OF COMPUTER ARCHITECTURES ON NUMERICAL ALGORITHMS

by

Garry Rodrigue
Lawrence Livermore National Laboratory
Livermore, California 94550

Workshop on High Speed Computing
Gleneden Beach, Oregon

March 31, 1981

Lawrence Livermore Laboratory

THE INFLUENCE OF COMPUTER ARCHITECTURES

ON

NUMERICAL ALGORITHMS

BY

GARRY RODRIGUE

COMPUTER RESEARCH GROUP

LAWRENCE LIVERMORE NATIONAL LABORATORY

IN THIS TALK,

1) AREAS THAT EFFECT NUMERICAL ALGORITHMS

2) HOW THESE AREAS EFFECT NUMERICAL ALGORITHMS

3) SHOW EXAMPLES OF HOW COMPUTERS HAVE INFLUENCED NUMERICAL ALGORITHMS BOTH IN THE PAST AND THE PRESENT

4) DISCUSS SOME OF THE DIFFICULTIES DEVELOPING NUMERICAL ALGORITHMS

AREAS THAT EFFECT NUMERICAL ALGORITHMS

THE NUMERICAL PROCESS IN SCIENCE

SCIENCE PROBLEM	NEUTRON TRANSPORT FLUID FLOW
↓	
MATH. MODEL	BOLTZMANN EQTN. NAVIER-STOKES EQTNS.
↓	
MATH. APPROX.	DIFFUSION INVISCID FLOW
↓	
NUM. APPROX.	BACKWARD EULER FORMULA LAX-WENDROFF FORMULA
↓	
NUM. ALG.	TRIDIAGONAL MATRIX SOLVER POISSON SOLVER
↓	
COMPUTER	CDC-7600 CRAY-1

HOW THESE AREAS INFLUENCE NUMERICAL ALGORITHM

NUMERICAL APPROXIMATIONS

PROVIDE DIRECTION

PROBLEM DEFINITION

NUMERICAL ALGORITHMS

PROVIDE DIRECTION

PROBLEM DEFINITION

HIGH SPEED COMPUTERS

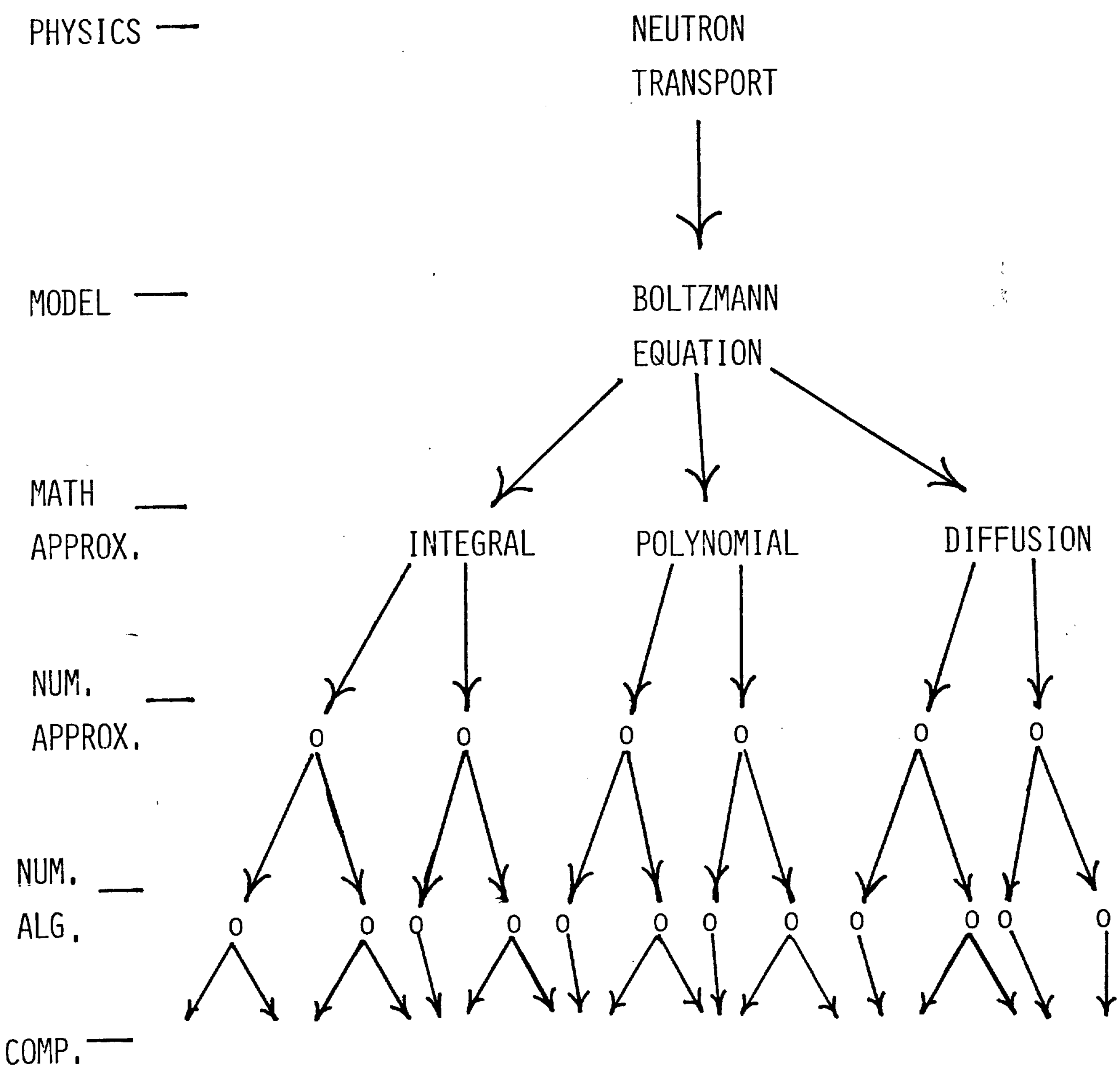
FOCUS ON ALGORITHMS FOR MATRIX CALCULATIONS

SPECIFICALLY:

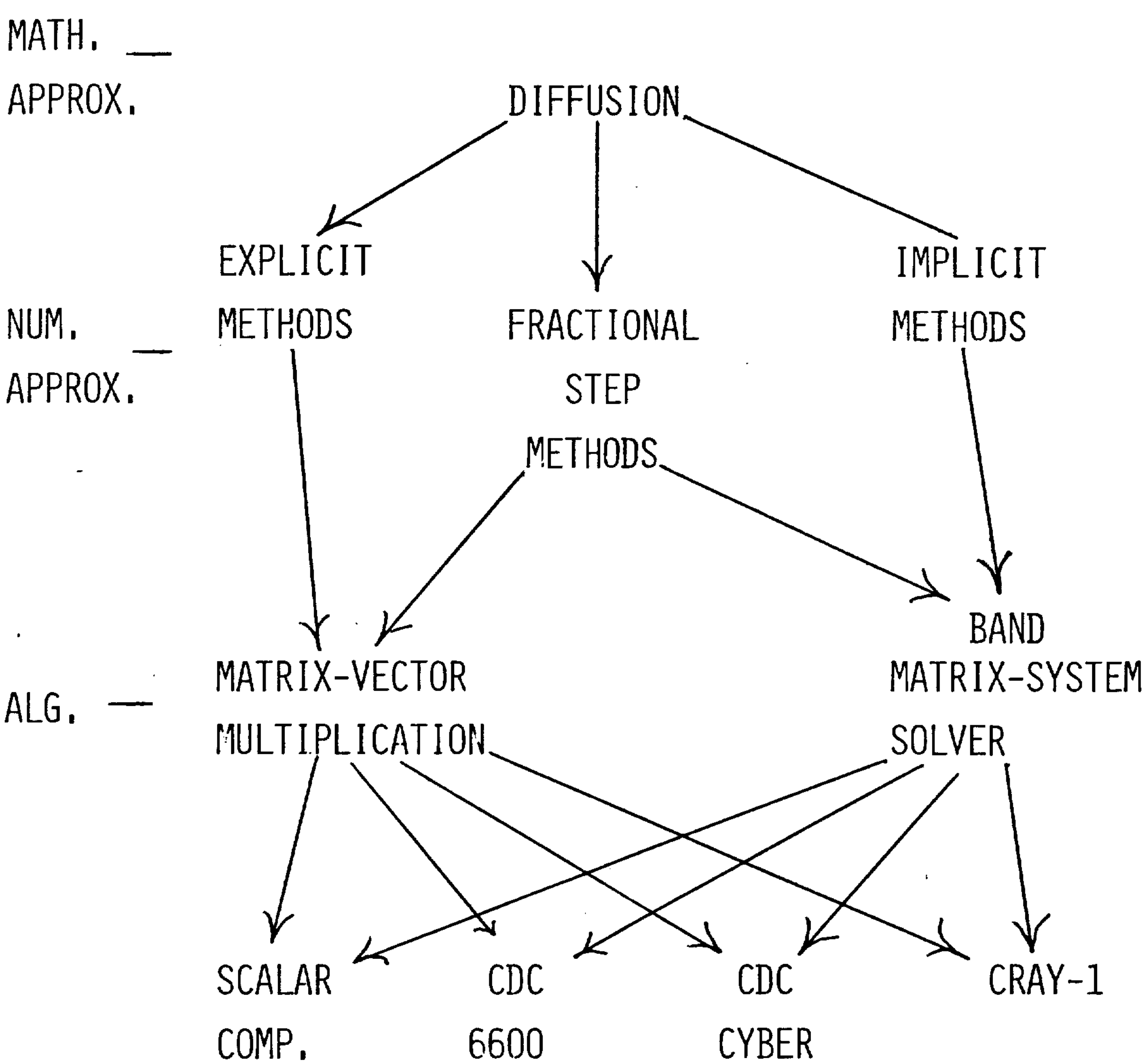
1) MATRIX-VECTOR MULTIPLICATION ALGORITHM

2) BANDED MATRIX INVERSION ALGORITHM

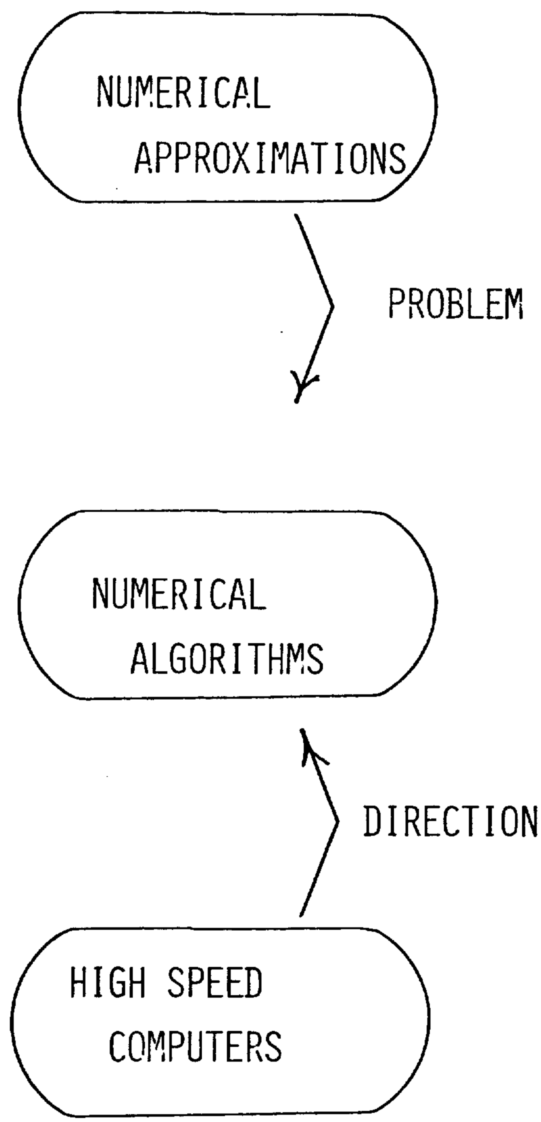
MATRIX ALGORITHMS ARE IMPORTANT FOR PERFORMING TRANSPORT CALCULATIONS:



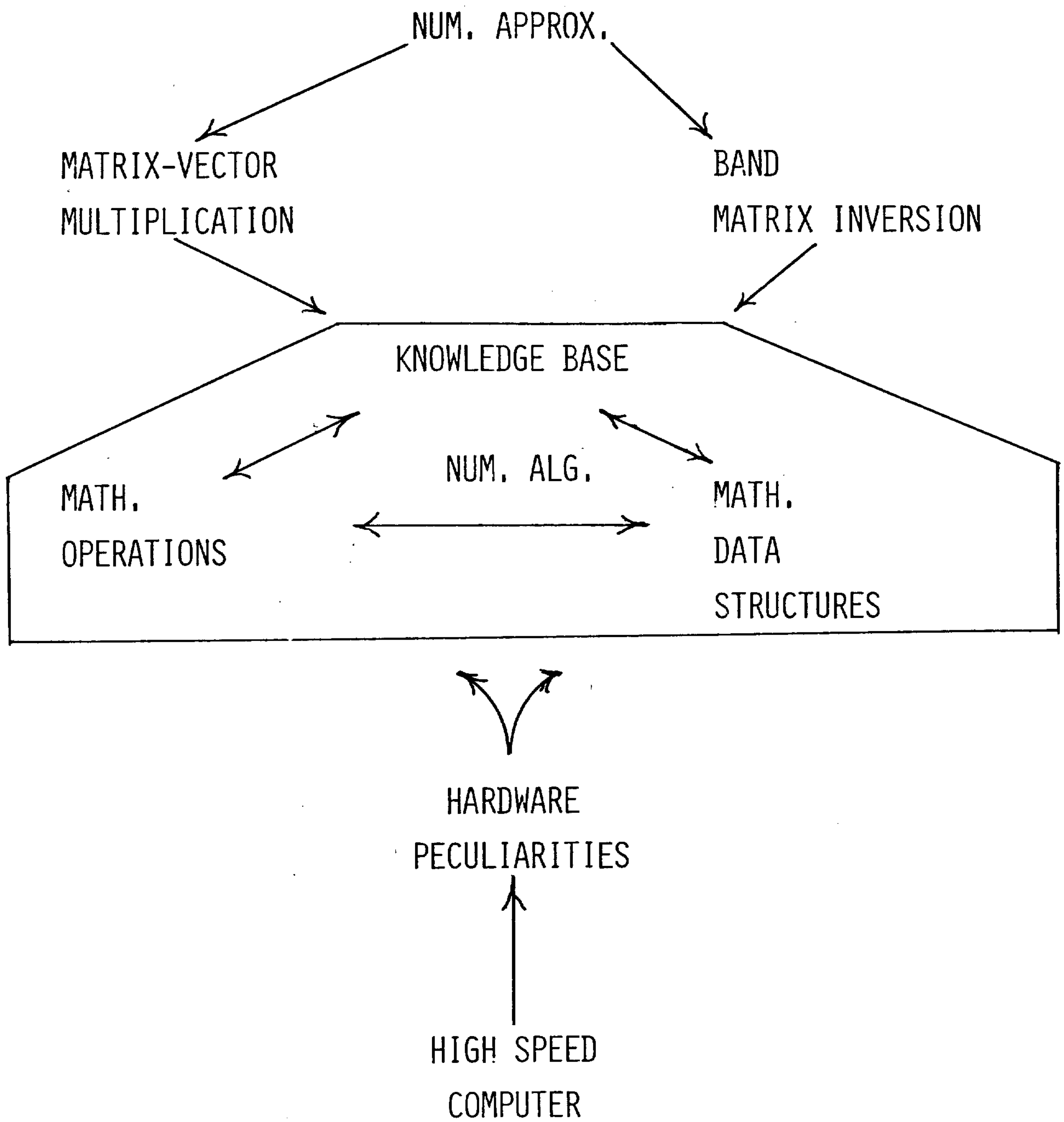
IN DIFFUSION CALCULATIONS,



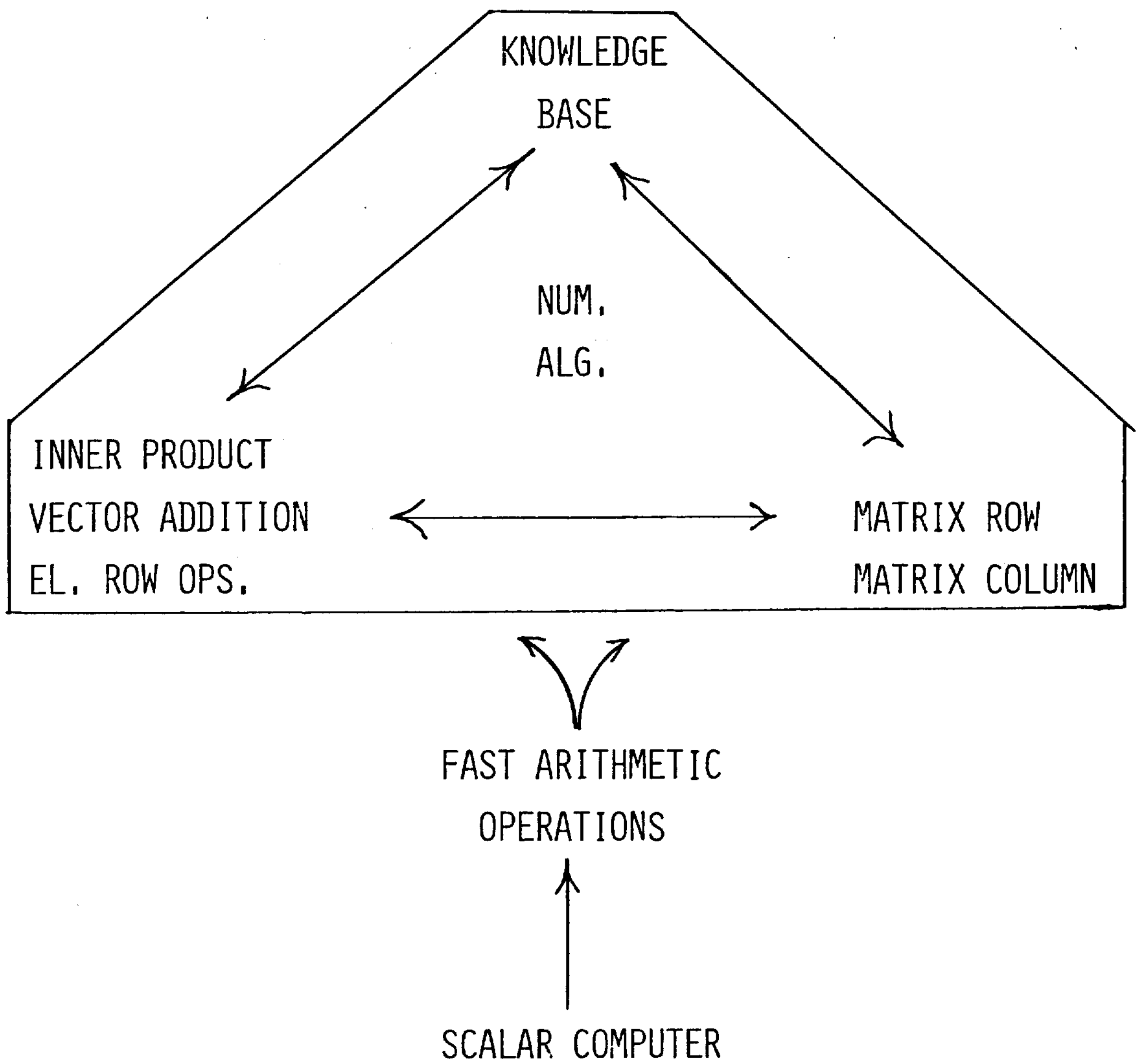
RECALL:


NUMERICAL APPROXIMATIONS
PROBLEM
NUMERICAL ALGORITHMS
DIRECTION
HIGH SPEED COMPUTERS

SPECIFICALLY,



AS AN EXAMPLE, LET US FIRST GO BACK TO THE "SCALAR ERA" TO SEE HOW MATRIX ALGORITHMS WERE DEVELOPED



EXAMPLE: MATRIX-VECTOR MULTIPLICATION

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NN} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} =$$

$$\begin{bmatrix} A_{11}X_1 + A_{12}X_2 + \cdots + A_{1N}X_N \\ A_{21}X_1 + A_{22}X_2 + \cdots + A_{2N}X_N \\ \vdots \\ A_{N1}X_1 + A_{N2}X_2 + \cdots + A_{NN}X_N \end{bmatrix}$$

MATRIX-VECTOR MULTIPLICATION ALGORITHMS

- INNER PRODUCT FORM

$$\begin{bmatrix} \text{———} \ \bar{A}_1 \ \text{———} \\ \text{———} \ \bar{A}_2 \ \text{———} \\ \vdots \\ \text{———} \ \bar{A}_N \ \text{———} \end{bmatrix} \begin{bmatrix} | \\ \bar{X} \\ | \end{bmatrix} = \begin{bmatrix} (\bar{A}_1, \bar{X}) \\ (\bar{A}_2, \bar{X}) \\ \vdots \\ (\bar{A}_N, \bar{X}) \end{bmatrix}$$

- OUTER PRODUCT FORM

$$\begin{bmatrix} | & | & & | \\ \bar{A}_1 & \bar{A}_2 & \text{— — —} & \bar{A}_N \\ | & | & & | \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} = \bar{A}_1 X_1 + \bar{A}_2 X_2 + \qquad + \bar{A}_N X_N$$

WITHOUT GOING INTO TOO MUCH DETAIL, THESE FORMS OF MATRIX MULTIPLICATION LED TO DEVELOPMENT OF MANY NEW ALGORITHMS.

1) CONJUGATE GRADIENT METHODS
2) HOUSEHOLDER METHODS
3) QR AND QZ METHODS
4) POWER AND DEFLATION METHODS
5) VARIABLE METRIC METHODS

ELEMENTARY ROW OPERATION

$$
\begin{array}{c} D \\ + \\ \\ \\ \end{array}
\begin{bmatrix}
A_{11} & A_{12} & \cdots & A_{1N} \\
A_{21} & A_{22} & \cdots & A_{2N} \\
\vdots & \vdots & & \vdots \\
A_{N1} & A_{N2} & \cdots & A_{NN}
\end{bmatrix}
\rightarrow
\begin{bmatrix}
A_{11} & & \cdots & A_{1N} \\
A_{21}+DA_{11} & A_{22}+DA_{12} & \cdots & A_{2N}+DA_{1N} \\
\vdots & & & \vdots \\
A_{N1} & & \cdots & A_{NN}
\end{bmatrix}
$$

- LED TO DEVELOPMENT OF

  1) VARIANTS OF GAUSS ELIMINATION ALGORITHM
  2) FACTORIZATION ALGORITHMS
  3) GIVENS ALGORITHM

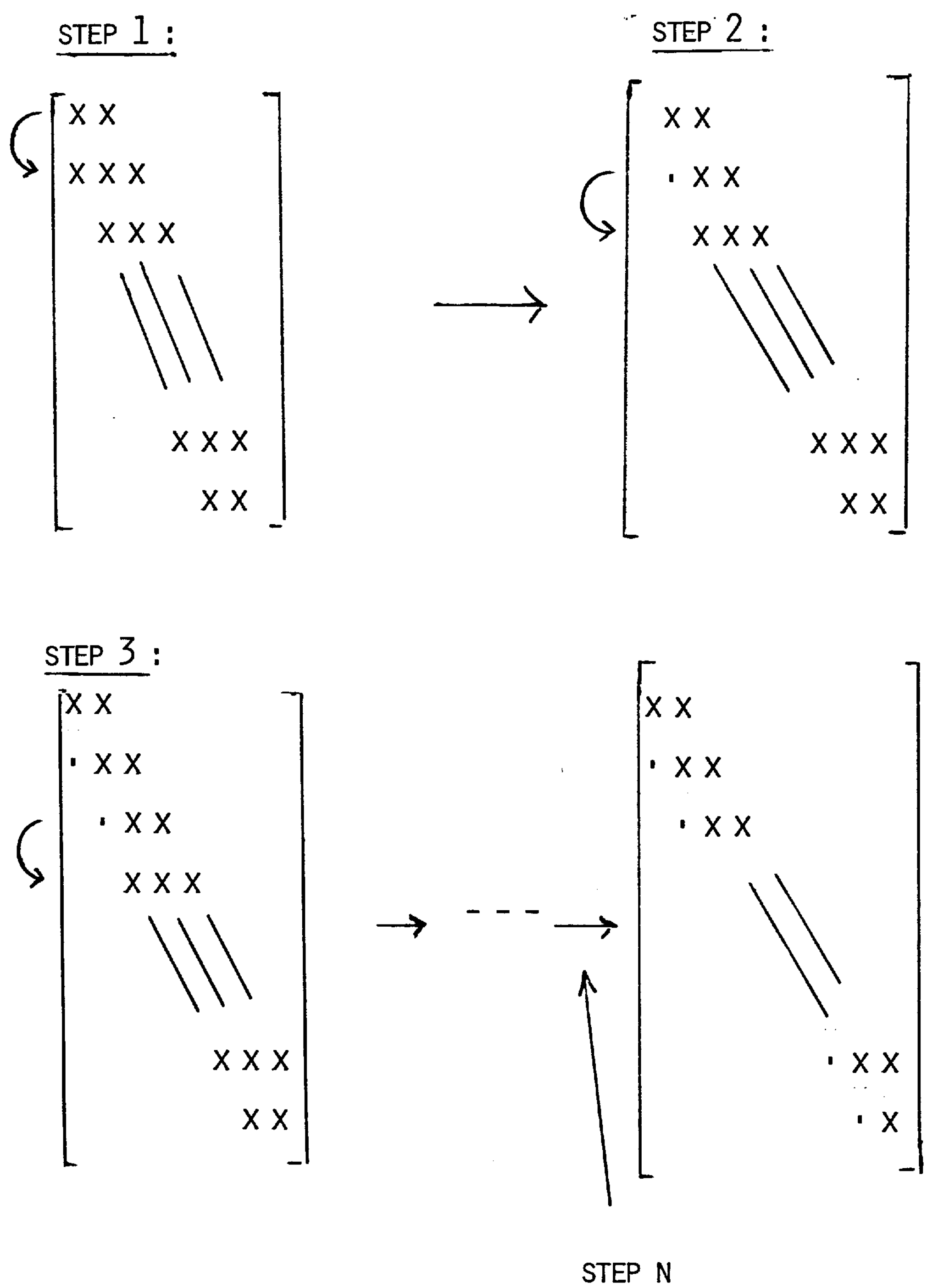
BANDED MATRICES

$$\begin{bmatrix} A_{11} & & A_{1K} & \\ & \ddots & & \ddots \\ & & & A_{KN} \\ A_{L1} & & \ddots & \\ & \ddots & & \\ & A_{NL} & & A_{NN} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix}$$

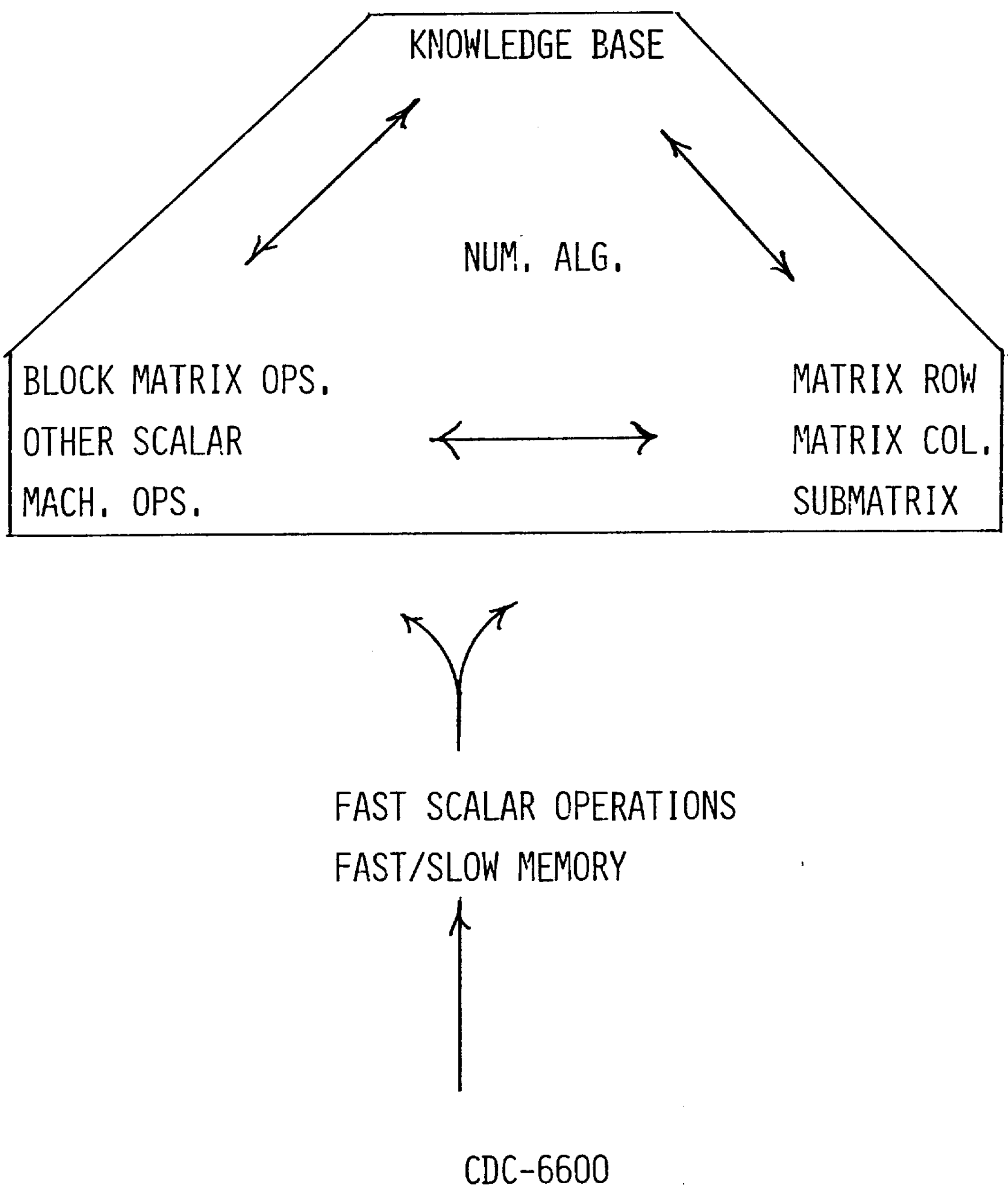
- FOR SCALAR MACHINES, ALGORITHMS DEVELOPED USING ROW AND COLUMN OPERATIONS WENT FASTER FOR BANDED SYSTEMS

THE CLASSICAL ALGORITHM IS GAUSSIAN ELIMINATION

- FORWARD SUBSTITUTION

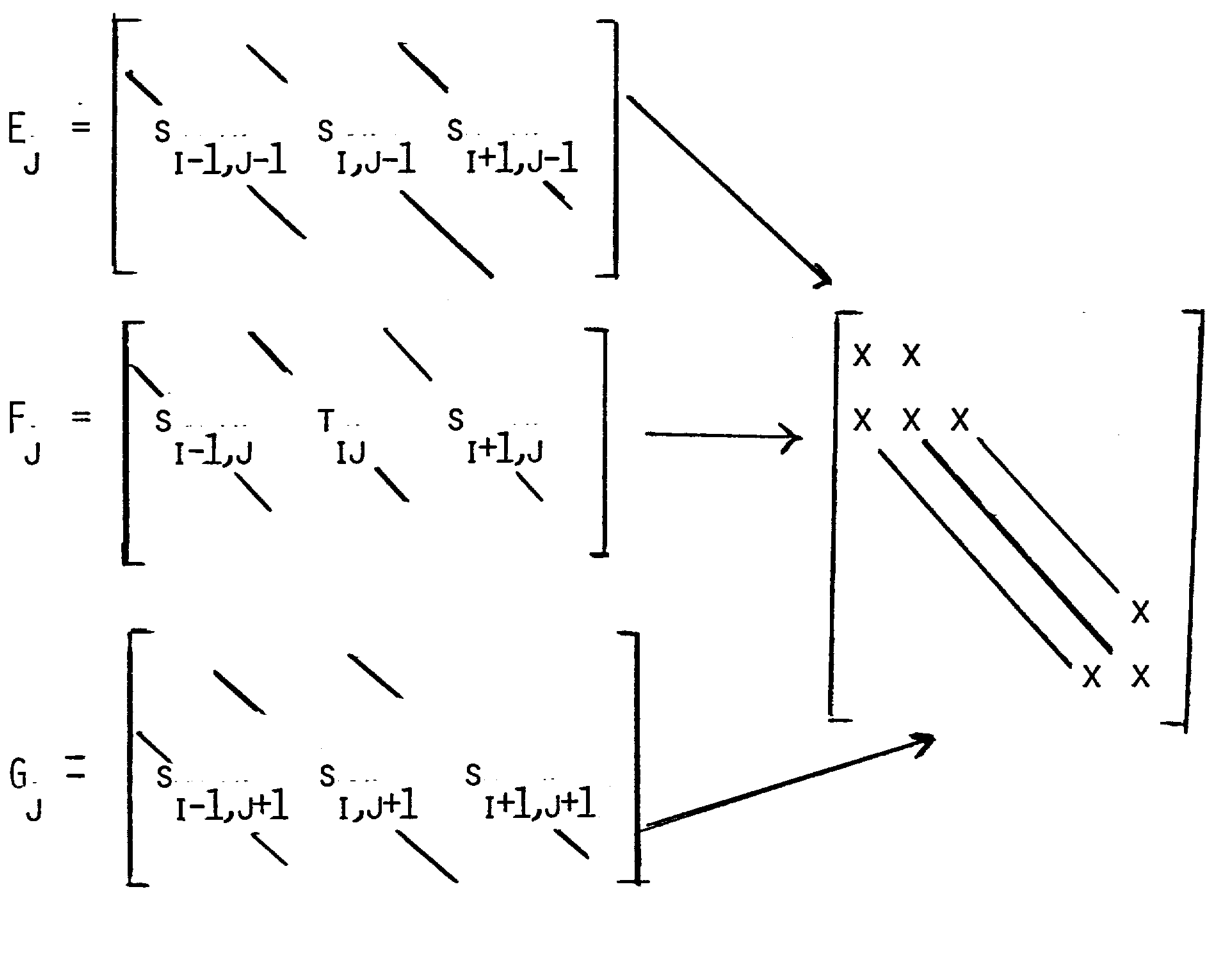


FAST/SLOW MEMORY MACHINES



A COMMON EXAMPLE OF A BLOCK-STRUCTURED MATRIX

$$
\begin{bmatrix}
F_1 & G_1 & & & \\
E_2 & F_2 & G_2 & & \\
 & \ddots & \ddots & \ddots & \\
 & & & & G_{L-1} \\
 & & & E_L & F_L
\end{bmatrix}
\begin{bmatrix}
\bar{U}_1 \\ \bar{U}_2 \\ \vdots \\ \bar{U}_L
\end{bmatrix}
$$



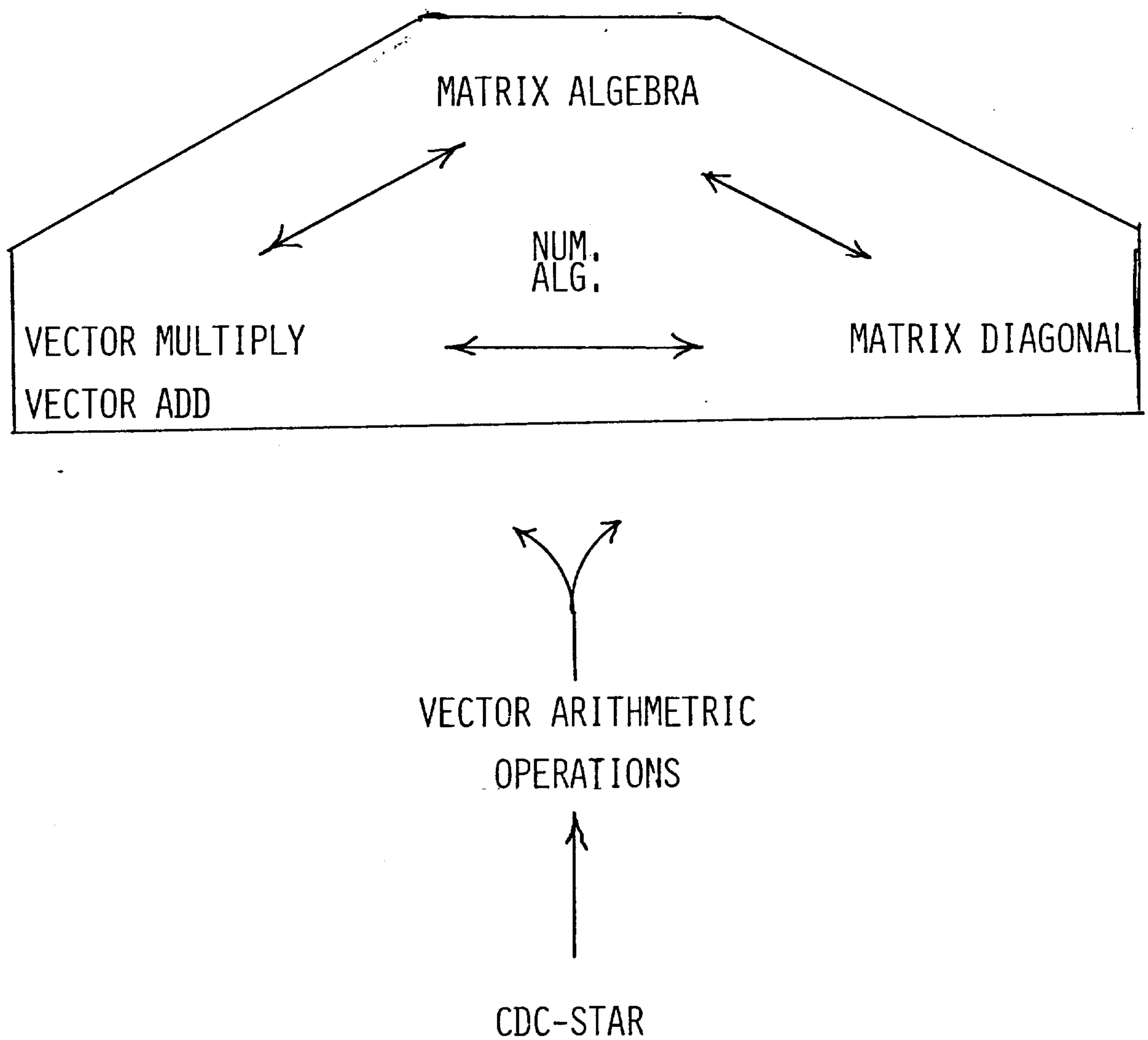
EFFICIENT ALGORITHMS WERE THOSE THAT COULD MANIPULATE THE BLOCKS OF THE MATRIX EFFECTIVELY

$$\begin{bmatrix} A_1 & B_1 & & & \\ C_2 & A_2 & B_2 & & \\ & \ddots & \ddots & \ddots & \\ & & & & B_{N-1} \\ & & & C_N & A_N \end{bmatrix}$$

A_I, B_I, C_I ARE MATRICES

- LED TO DEVELOPMENT OF
  1) BLOCK ITERATIVE METHODS
  2) FAST POISSON/BIHARMONIC SOLVERS
  3) COMPANION MATRIX ALGORITHMS

CDC-STAR


MATRIX ALGEBRA
NUM.
ALG.
VECTOR MULTIPLY
VECTOR ADD
MATRIX DIAGONAL
VECTOR ARITHMETIC
OPERATIONS
CDC-STAR

CONSTRAINTS OF THE STAR

1) SLOW SCALAR ARITHMETIC OPERATIONS

2) SLOW VECTOR INSTRUCTIONS WHEN VECTORS ARE SHORT

NEW PAWNS IN THE GAME: VECTOR ARITHMETIC INSTRUCTIONS

1) SCALAR X VECTOR

$$A \begin{bmatrix} X_1 \\ X_2 \\ | \\ | \\ | \\ | \\ X_N \end{bmatrix} = \begin{bmatrix} AX_1 \\ AX_2 \\ | \\ | \\ | \\ | \\ AX_N \end{bmatrix}$$

2) VECTOR ADDITION

$$\begin{bmatrix} X_1 \\ X_2 \\ | \\ | \\ | \\ | \\ | \\ X_N \end{bmatrix} + \begin{bmatrix} Y_1 \\ Y_2 \\ | \\ | \\ | \\ | \\ | \\ Y_N \end{bmatrix} = \begin{bmatrix} X_1 + Y_1 \\ X_2 + Y_2 \\ | \\ | \\ | \\ | \\ | \\ X_N + Y_N \end{bmatrix}$$

3) VECTOR MULTIPLICATION**

$$\begin{bmatrix} X_1 \\ X_2 \\ | \\ | \\ | \\ X_N \end{bmatrix} \cdot \begin{bmatrix} Y_1 \\ Y_2 \\ | \\ | \\ | \\ Y_N \end{bmatrix} = \begin{bmatrix} X_1Y_1 \\ X_2Y_2 \\ | \\ | \\ | \\ X_NY_N \end{bmatrix}$$

MATRIX-VECTOR MULTIPLICATION

(1)

$$\begin{bmatrix} \text{———} \ \bar{A}_1 \ \text{———} \\ \text{———} \ \bar{A}_2 \ \text{———} \\ \vdots \\ \text{———} \ \bar{A}_N \ \text{———} \end{bmatrix} \begin{bmatrix} | \\ \bar{X} \\ | \end{bmatrix} = \begin{bmatrix} (\bar{A}_1, \bar{X}) \\ (\bar{A}_2, \bar{X}) \\ \vdots \\ (\bar{A}_N, \bar{X}) \end{bmatrix}$$

(2)

$$\begin{bmatrix} | & | & & | \\ \bar{A}_1 & \bar{A}_2 & \cdots & \bar{A}_N \\ | & | & & | \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} = \bar{A}_1 X_1 + \cdots + \bar{A}_N X_N$$

WHAT HAPPENS ON A BANDED MATRIX?

$$
\begin{bmatrix}
A_{11} & & A_{1K} & & \\
& \ddots & & \ddots & \\
& & & & A_{KN} \\
A_{L1} & & \ddots & & \\
& \ddots & & \ddots & \\
& & A_{NL} & & A_{NN}
\end{bmatrix}
\begin{bmatrix}
X_1 \\ X_2 \\ \vdots \\ X_N
\end{bmatrix}
$$

$$
= \bar{A}_1 X_1 + \bar{A}_2 X_2 + \qquad + \bar{A}_N X_N
$$

$$
\begin{bmatrix}
A_{11} \\ 0 \\ \vdots \\ 0 \\ A_{L1} \\ 0 \\ \vdots \\ 0
\end{bmatrix}
\cdot X_1
$$

BETTER OFF USING SCALAR ARITHMETIC INSTRUCTIONS

WHY COULDN'T THE SUBMATRIX DATA-STRUCTURE BE USED?

$$
\begin{bmatrix}
A_1 & B_1 & & \\
C_2 & A_2 & B_2 & \\
 & \ddots & \ddots & \ddots \\
 & & & B_{N-1} \\
 & & C_N & A_N
\end{bmatrix}
$$

- ALGORITHMS DERIVED USING BLOCK STRUCTURES RESULTED IN SHORT VECTORS

$$
\begin{bmatrix}
A_1 & & & \\
 & A_2 & & \\
 & & A_3 & \\
 & & & \ddots & \\
 & & & & A_N
\end{bmatrix}
$$

TAKE ANOTHER LOOK

$$\begin{bmatrix} A_{11} & & A_{1K} & & \\ & \ddots & & \ddots & \\ & & \ddots & & A_{KN} \\ A_{L1} & & & \ddots & \\ & \ddots & & & \ddots \\ & & A_{NL} & & A_{1N} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ \vdots \\ X_N \end{bmatrix}$$

$$= \left[\begin{array}{lllllll} & & & A_{11}\ X_1 & + & A_{1K} & X_K \\ & & & A_{22}\ X_2 & + & A_{2,K+1} & X_{K+1} \\ & & & & \vdots & & \\ A_{L1} & X_1 & + & A_{LL}\ X_L & + & A_{L,K+L} & X_{L+L} \\ & & \vdots & & \vdots & & \\ A_{K,K-L} & X_{K-L} & + & A_{KK}\ X_K & + & A_{KN} & X_N \\ & & \vdots & & & & \\ A_{NL} & X_L & + & A_{NN}\ X_N & & & \end{array}\right]$$

IN VECTOR NOTATION

$$\begin{bmatrix} \ddots & & \bar{A}_K & \\ & \bar{A}_0 & & \ddots \\ \bar{A}_{-L} & & \ddots & \\ & \ddots & & \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix}$$

$$\equiv \bar{A}_0 \cdot \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} + \bar{A}_{-L} \cdot \begin{bmatrix} X_1 \\ \vdots \\ X_L \end{bmatrix} + \bar{A}_K \cdot \begin{bmatrix} X_K \\ \vdots \\ X_N \end{bmatrix}$$

- GENERALIZATION TO MORE COMPLICATED SYSTEMS IS EASY

GENERAL COMMENTS

- DIAGONAL ALGORITHM IS

  30 - 40 TIMES FASTER ON STAR-100
  5 - 8 TIMES FASTER ON CRAY-1

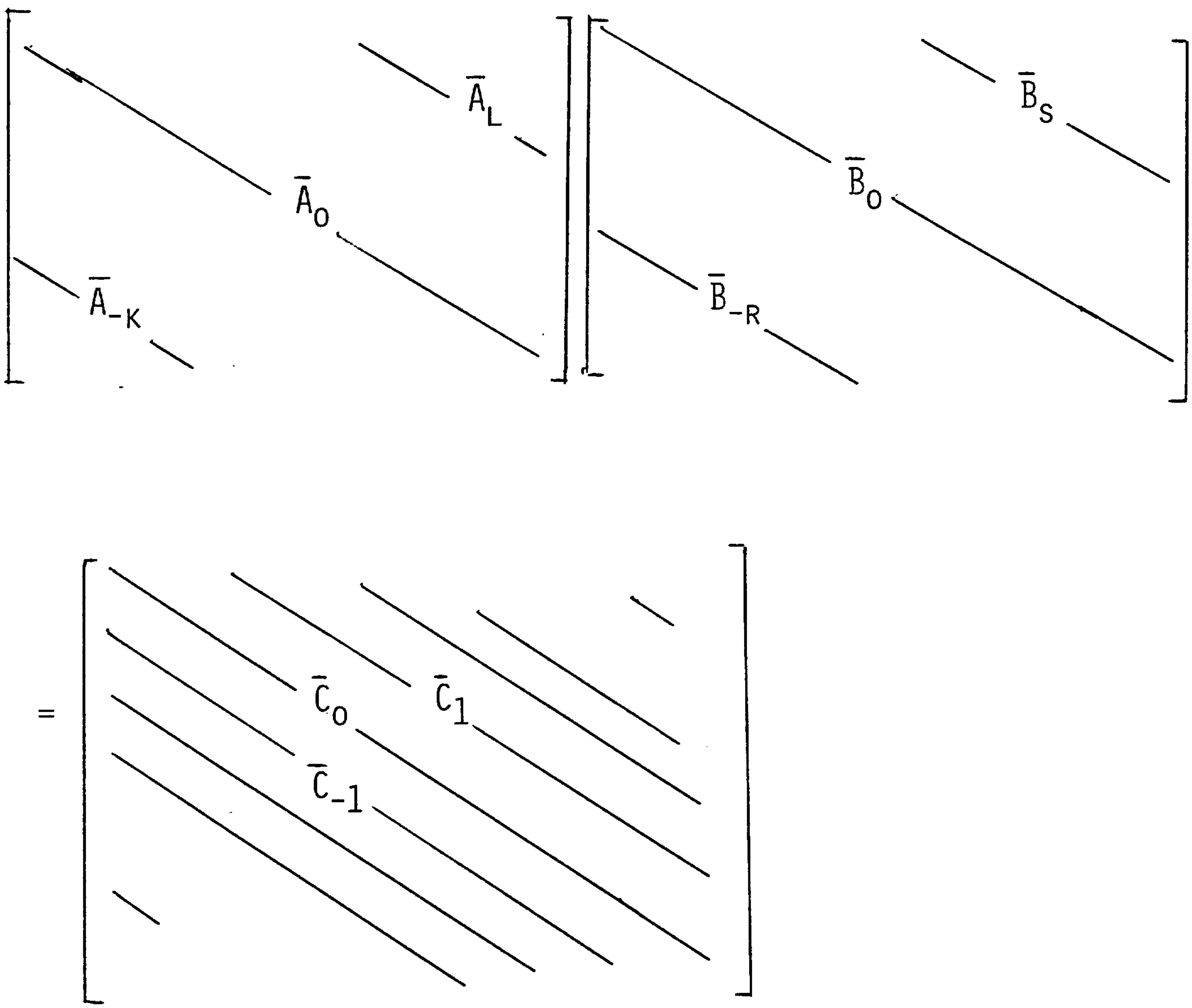
  THAN CLASSICAL ROW AND COLUMN ALGORITHM

- NO MATHEMATICAL OR GEOMETRICAL THEORY OF MATRIX-VECTOR ALGORITHM USING DIAGONALS AS A VECTOR STRUCTURE HAS EVER BEEN CARRIED OUT.

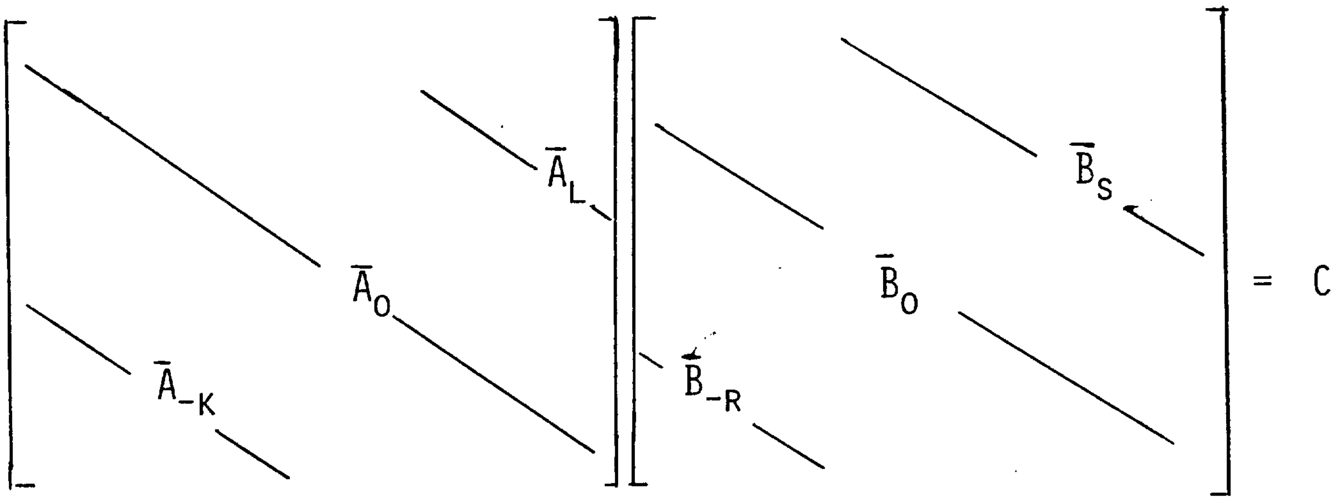
- WE NEED THIS IF MORE EFFICIENT ALGORITHMS FOR VECTOR OR MULTI-PROCESSING MACHINES ARE TO BE INVENTED.

AN EXAMPLE OF HOW RESULTS USING DIAGONAL VECTORS CAN LEAD TO NEW ALGORITHMS

- MATRIX MULTIPLICATION: AB = C



<u>RESULT 1:</u> IF DIAGONALS $\bar{A}_I$ AND $\bar{B}_J$ ARE NON-ZERO, THEN DIAGONAL $\bar{C}_{I+J}$ IS NON-ZERO



- <u>NON-ZERO DIAGONALS OF C</u>

$\bar{C}_O$

$\bar{C}_{-K-R}$

$\bar{C}_{-K+S}$

$\bar{C}_{L-R}$

$\bar{C}_{L+S}$

$\bar{C}_{-K}$

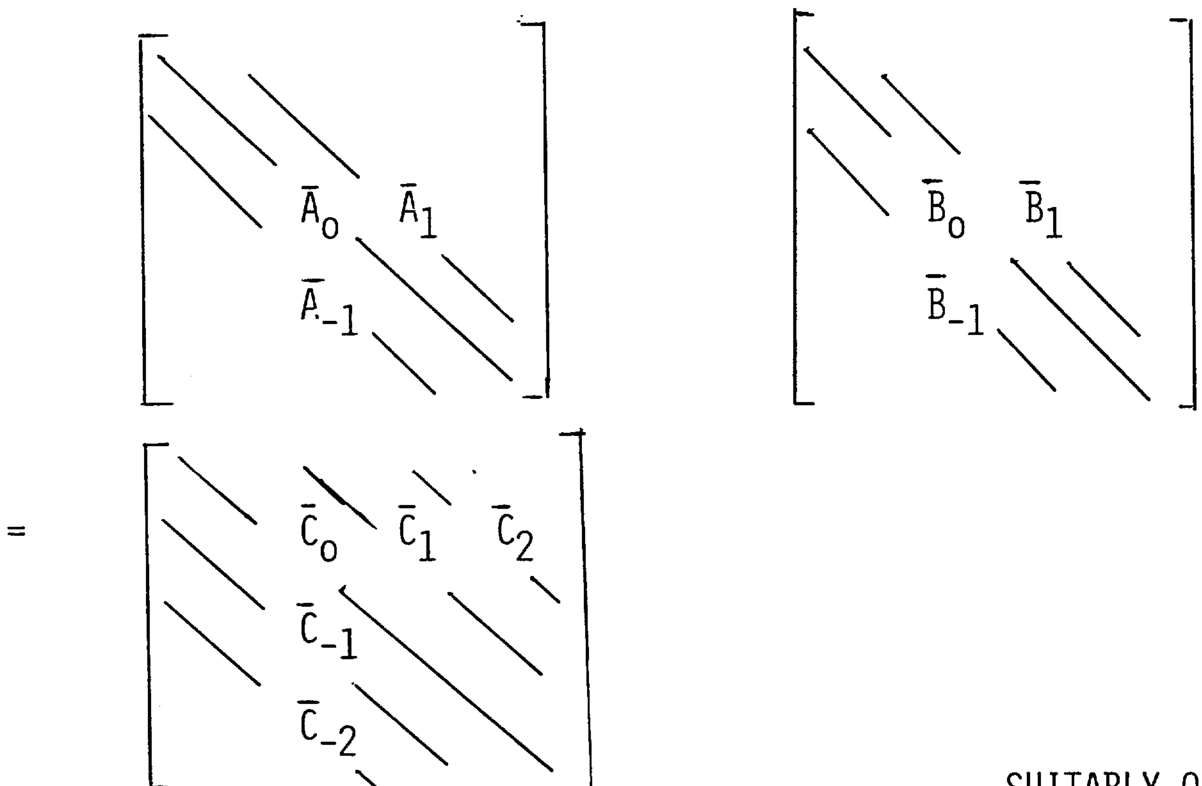
$\bar{C}_L$

$\bar{C}_{-R}$

$\bar{C}_S$

RESULT 2: THE PRODUCT $\bar{A}_I \quad \bar{B}_J$ CONTRIBUTES TO THE FORMATION OF $\bar{C}_{I+J}$

EXAMPLE



SUITABLY OFFSET

$$\bar{C}_0 = \bar{A}_0 \bar{B}_0 + \bar{A}_{-1} \bar{B}_1 + \bar{A}_1 \bar{B}_{-1}$$

I.E., EACH DIAGONAL CAN BE CONSTRUCTED FROM VECTOR PRODUCTS AND ADDITIONS

WE USE THESE TWO RESULTS TO GENERATE AN ALGORITHM TO SOLVE A TRIDIAGONAL SYSTEM OF EQUATIONS

PROBLEM STATEMENT: $TX = B$

$$\begin{bmatrix} A_{11} & A_{12} & & \\ A_{21} & A_{22} & A_{23} & \\ & \ddots & \ddots & \ddots \\ & & & A_{N-1,N} \\ & & A_{N1} & A_{NN} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix}$$

DIAGONAL FORM

$$\begin{bmatrix} \ddots & \ddots & & \\ \ddots & \bar{A}_0 & \bar{A}_1 & \\ & \bar{A}_{-1} & \ddots & \ddots \\ & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \bar{X} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \bar{B} \\ \vdots \end{bmatrix}$$

STEP 1

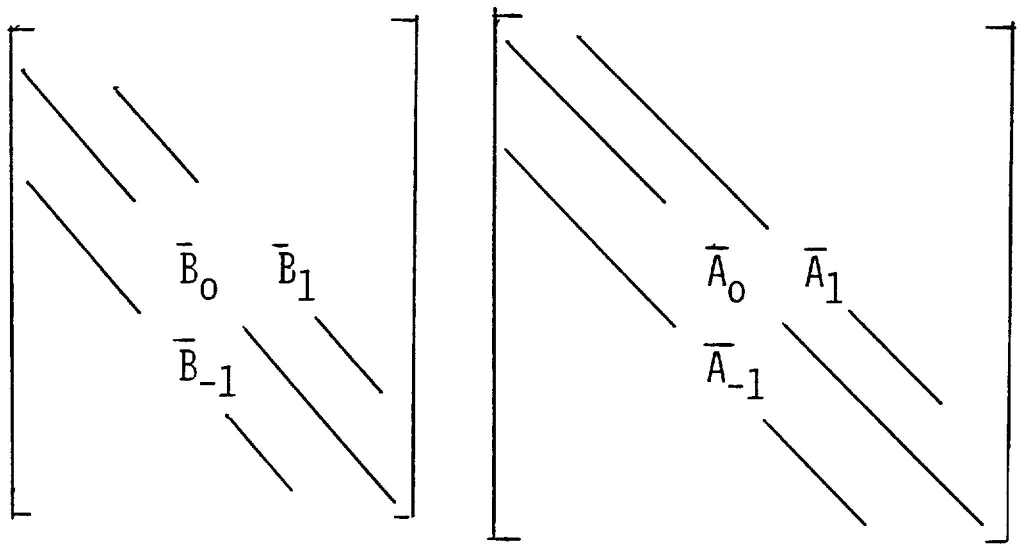
CONSTRUCT TRIDIAGONAL MATRIX Q_1 SO THAT: $Q_1\ T =$

$$\begin{bmatrix} \ddots & \ddots & & \\ \ddots & \bar{B}_0 & \bar{B}_1 & \\ & \bar{B}_{-1} & \ddots & \ddots \\ & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} \ddots & \ddots & & \\ \ddots & \bar{A}_0 & \bar{A}_1 & \\ & \bar{A}_{-1} & \ddots & \ddots \\ & & \ddots & \ddots \end{bmatrix}$$

$$= \begin{bmatrix} \ddots & & \ddots & & \\ & \bar{C}_0 & & \bar{C}_2 & \\ \ddots & & \ddots & & \ddots \\ & \bar{C}_{-2} & & \ddots & \\ & & \ddots & & \ddots \end{bmatrix}, \quad \text{I.E. } \bar{C}_{-1} = \bar{C}_1 = 0$$

AND $\bar{B}_0 = 1$

PROCEDURE



ANALYSIS: FROM RESULTS 1 AND 2,

$$\bar{B}_o \bar{A}_1 + \bar{B}_1 \bar{A}_o = \bar{C}_1 = 0$$

$$\bar{B}_o \bar{A}_{-1} + \bar{B}_{-1} \bar{A}_o = \bar{C}_{-1} = 0$$

BUT $\bar{B}_o = 1$:

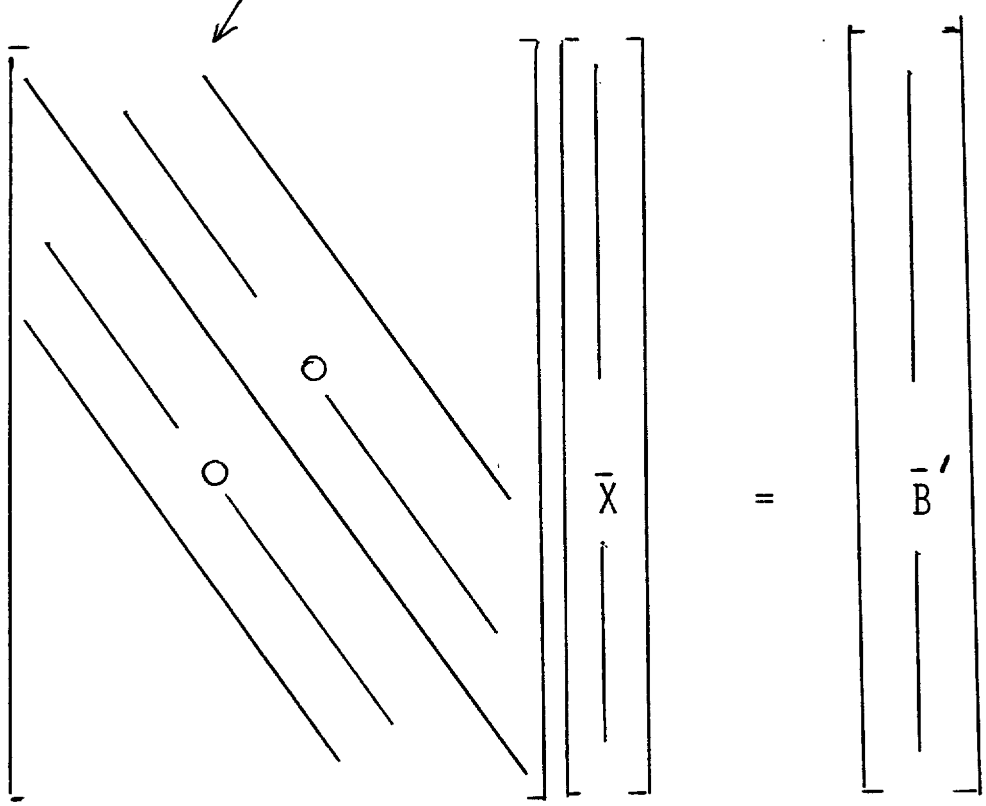
$$\bar{B}_1 = - \bar{A}_1/\bar{A}_o$$

$$\bar{B}_{-1} = - \bar{A}_{-1}/\bar{A}_o$$

WHAT HAS HAPPENED

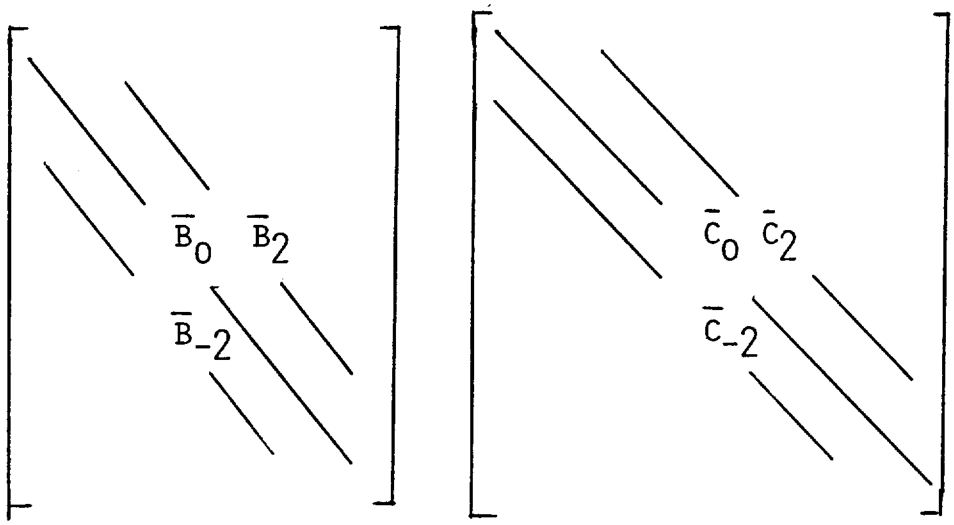
$$\underbrace{Q_1 \; A \; X} \; = \; \underbrace{Q_1 \; B}$$

VECTOR PROCESS

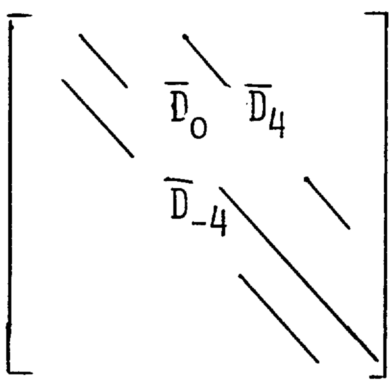


STEP 2: CONSTRUCT MATRIX Q_2 SO THAT

$Q_2 \, Q_1 \, A =$



$=$

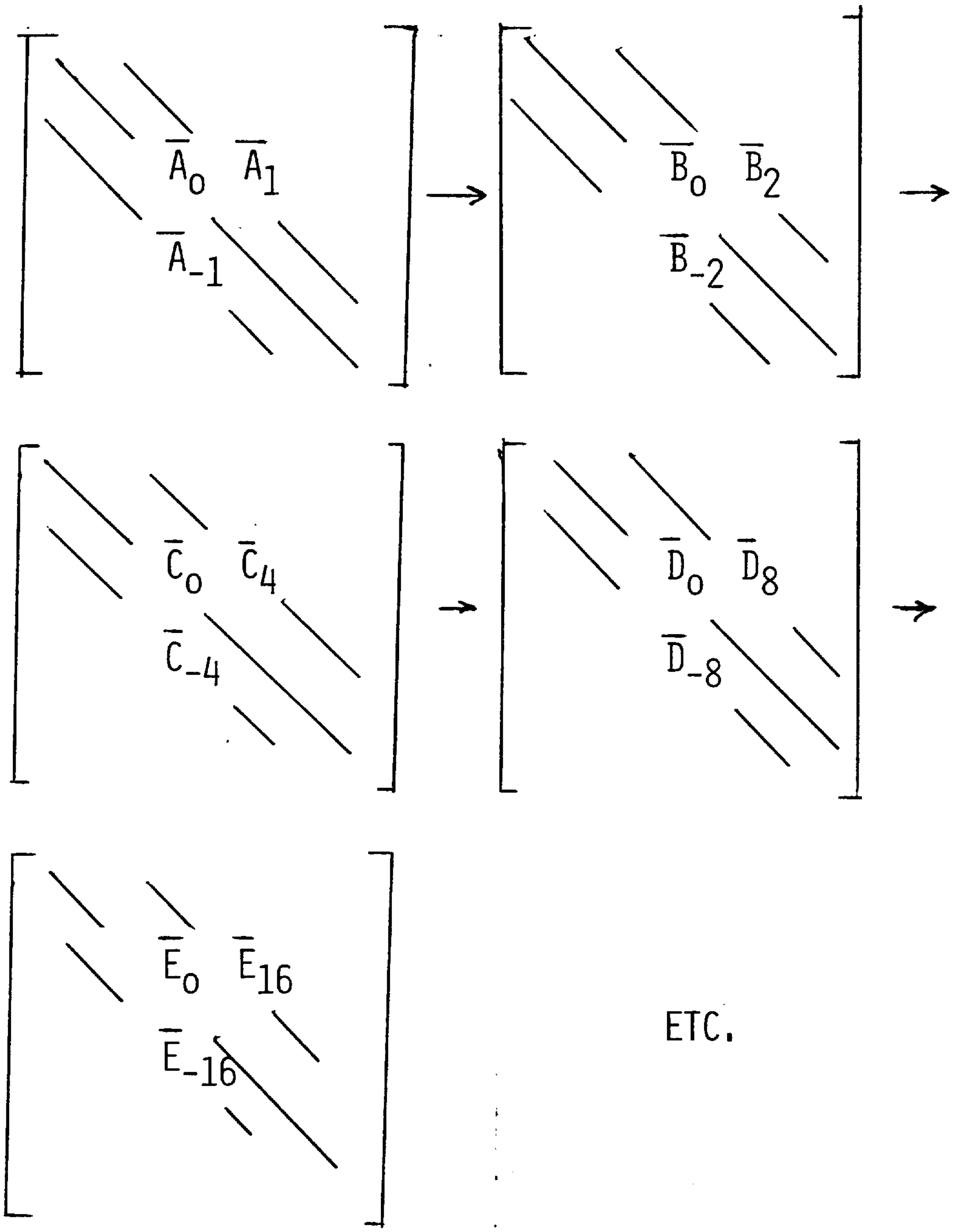


$, \quad \bar{B}_0 = 1$

- CONSTRUCT Q_2 AS BEFORE: $\bar{B}_{-2} = -\bar{A}_{-2}/\bar{A}_0$

$$\bar{B}_2 = -\bar{A}_2/\bar{A}_0$$

STEP-BY-STEP PROCESS



OTHER VECTOR ALGORITHMS CAN BE GENERATED WITH THIS PROCEDURE

- RECURSIVE - DOUBLING
- ODD-EVEN REDUCTION
- PARALLEL - GAUSS ELIMINATION
- CYCLIC REDUCTION**

AFTER $\log_2 N = M$ STEPS:

$$Q_M Q_{M-1} \cdot \cdot \cdot Q_1 A\bar{X} = \begin{bmatrix} \ddots & & \\ & \bar{Z}_o & \\ & & \ddots \end{bmatrix} \begin{bmatrix} | \\ \bar{X} \\ | \end{bmatrix}$$

$$= Q_M Q_{M-1} \cdot \cdot \cdot Q_1 \bar{B}$$

$$= \bar{B}'$$

FINAL SOLUTION:

$$\bar{X} = \bar{B}' / \bar{Z}_o$$

CYCLIC REDUCTION vs. RECURSIVE DOUBLING

STEPS

BOTH REQUIRE LOG_2N STEPS

AVERAGE VECTOR LENGTHS

RECURSIVE DOUBLING: $N - 2^{K-1}$

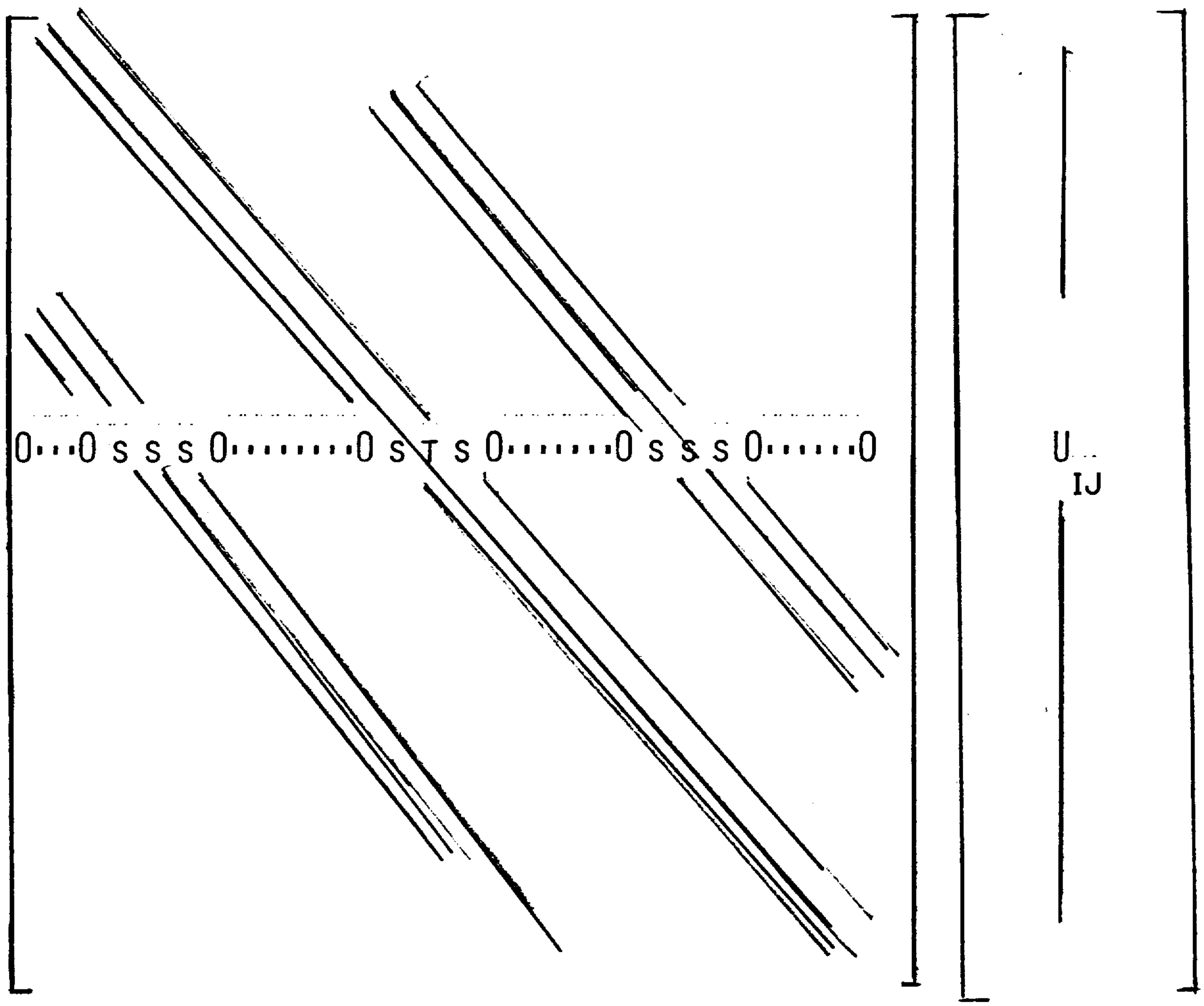
CYCLIC REDUCTION: $N/2^K$

COMPLEXITY

RECURSIVE DOUBLING: EASY TO CODE, EASY TO UNDERSTAND

CYCLIC REDUCTION: DIFFICULT TO CODE, DIFFICULT TO UNDERSTAND

A COMMON EXAMPLE OF A DIAGONALLY STRUCTURAL MATRIX



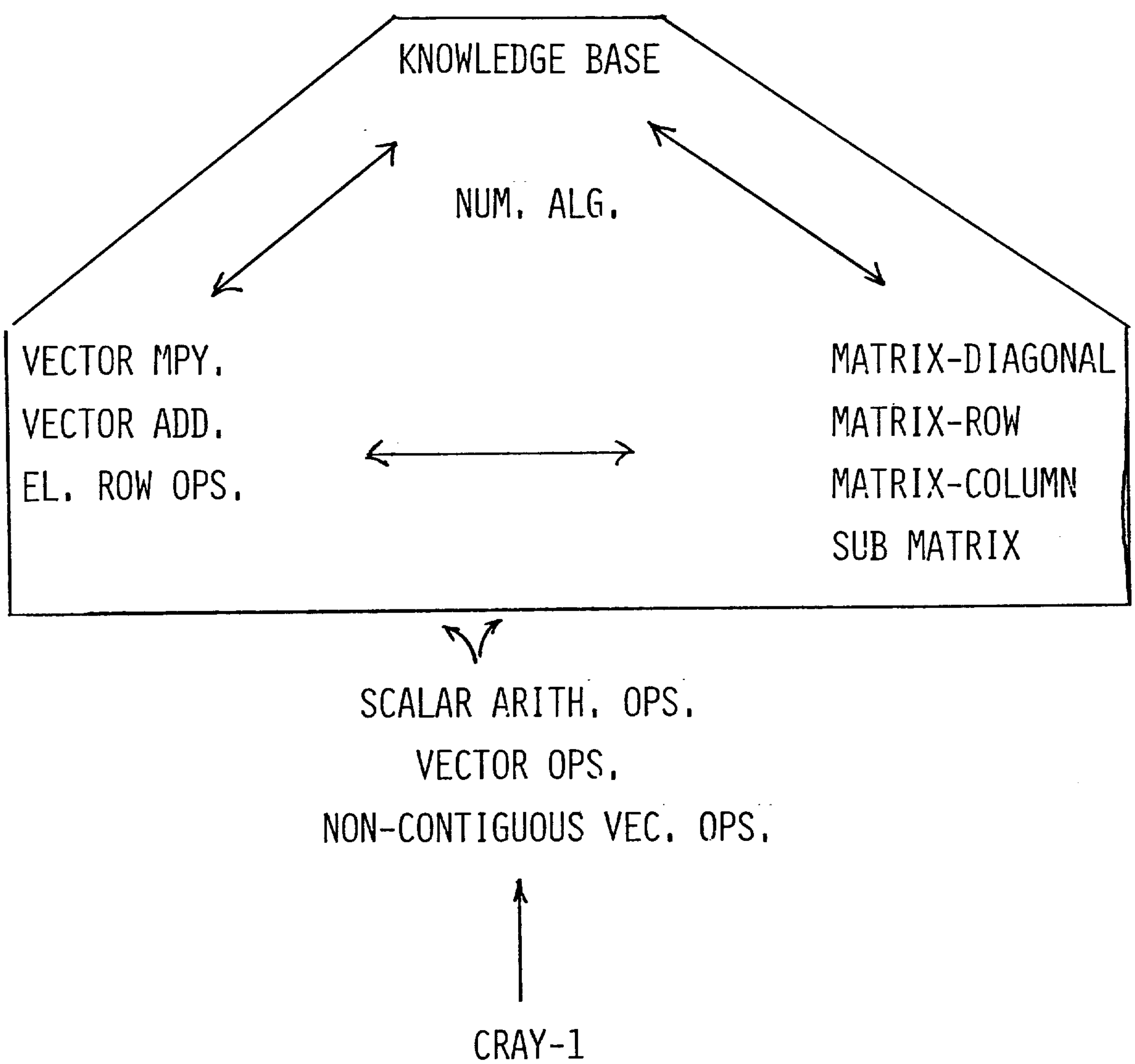
OPTIMAL ALGORITHMS FOR VECTOR PROCESSORS

- OPTIMAL ALGORITHMS WILL USE
  - COLUMN VECTOR OPERATIONS
  - ROW VECTOR OPERATIONS
  - DIAGONAL VECTOR OPERATIONS

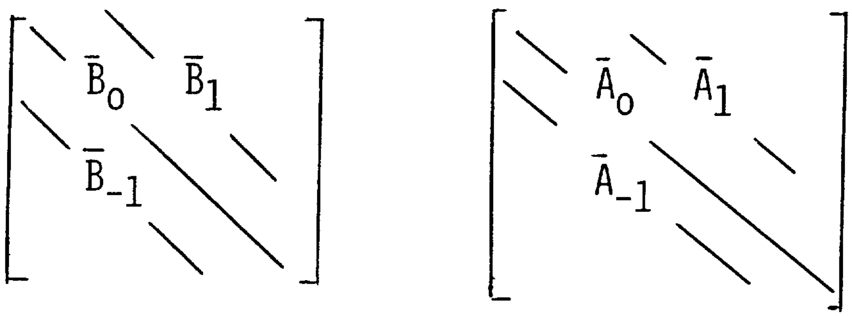
  IN AN EFFICIENT MANNER

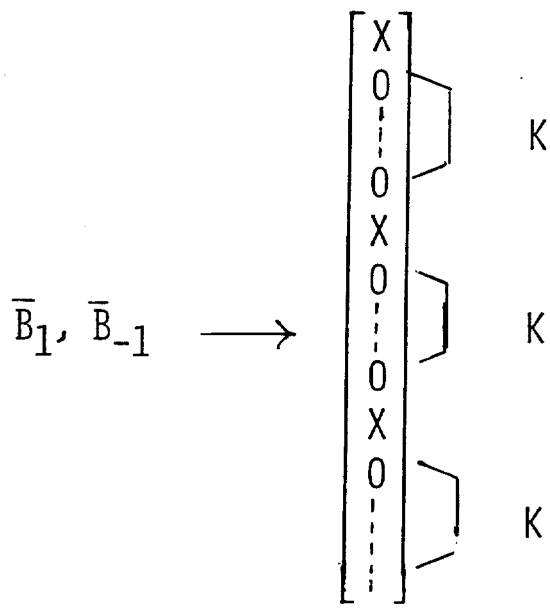
- EG. WHEN VECTORS (I.E. DIAGONALS) BECOME TOO SHORT, SWITCH TO ROW OR COLUMN VECTOR OPERATIONS TO CREATE ALGORITHM

CRAY-1

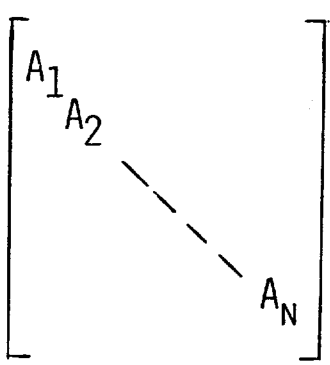

KNOWLEDGE BASE
NUM. ALG.
VECTOR MPY.
VECTOR ADD.
EL. ROW OPS.
MATRIX-DIAGONAL
MATRIX-ROW
MATRIX-COLUMN
SUB MATRIX
SCALAR ARITH. OPS.
VECTOR OPS.
NON-CONTIGUOUS VEC. OPS.
CRAY-1

EFFICIENT MATRIX-MULTIPLICATION ALGORITHMS BY NON-CONTIGUOUS DIAGONAL VECTORS (EVENLY INCREMENTED) WAS NOW POSSIBLE





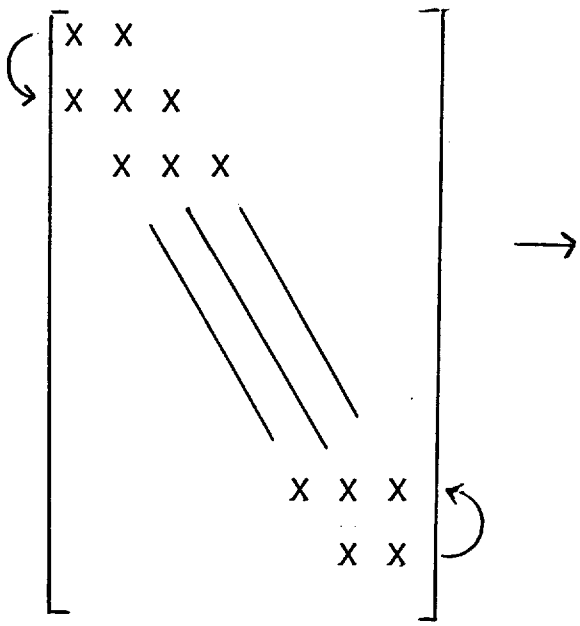
- NOW CAN HANDLE



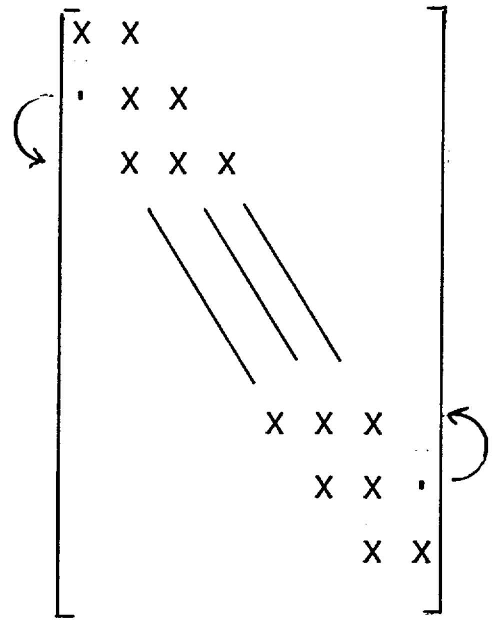
- LEADS TO VARIANTS OF CYCLIE REDUCTION ALG.

THE FOLDING ALGORITHM (FORWARD SUBSTITUTION)

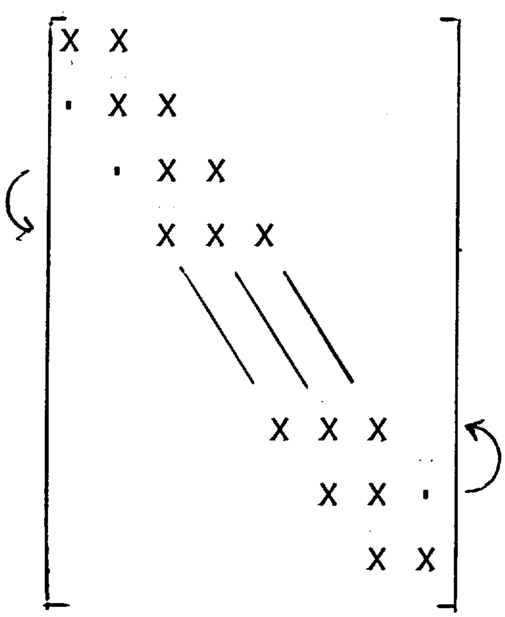
STEP 1:



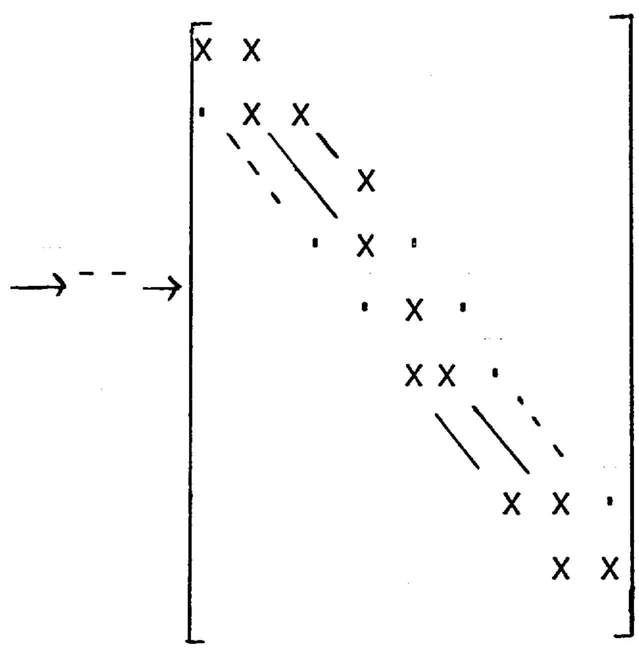
STEP 2:



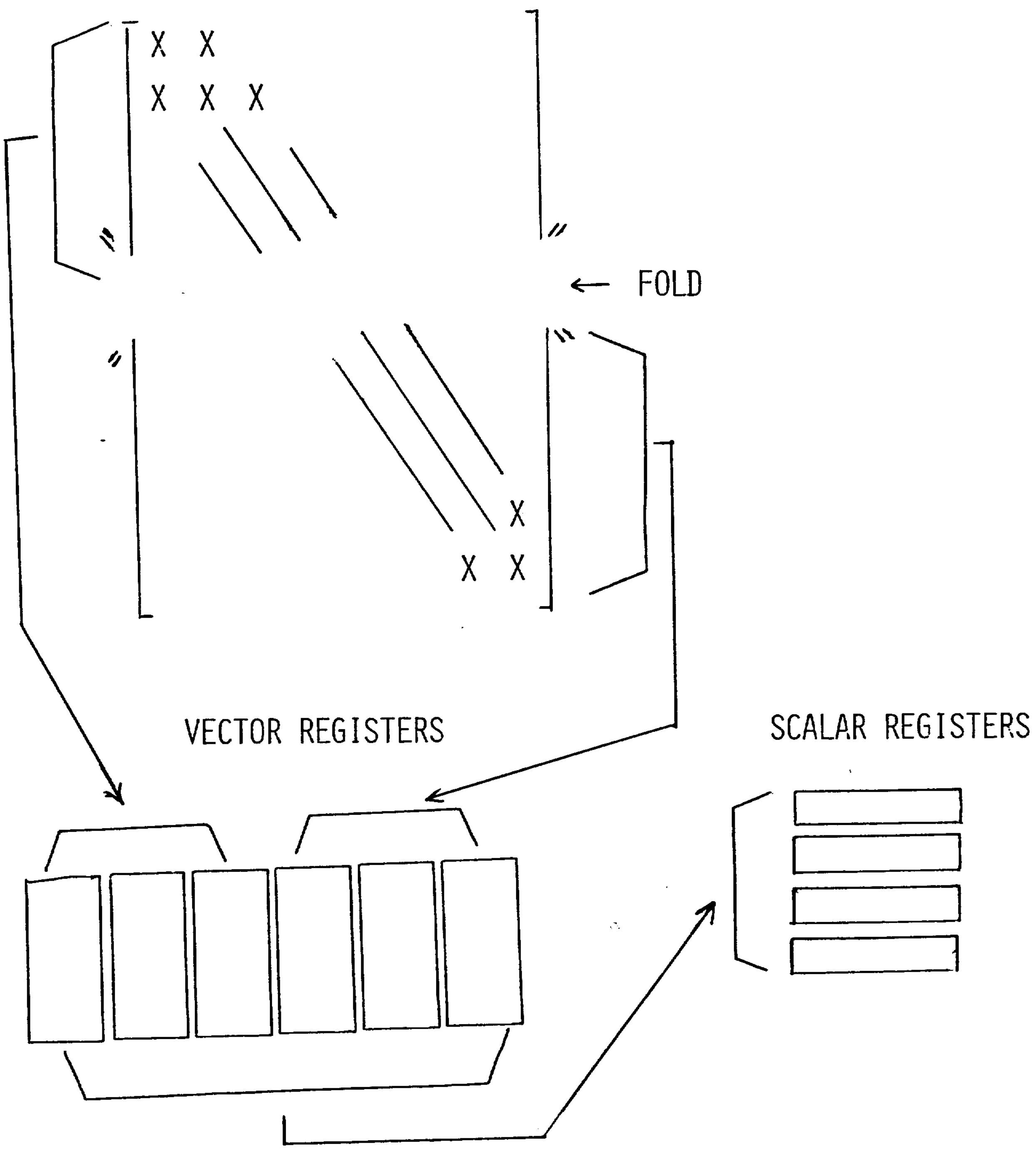
STEP 3:



STEP N/2



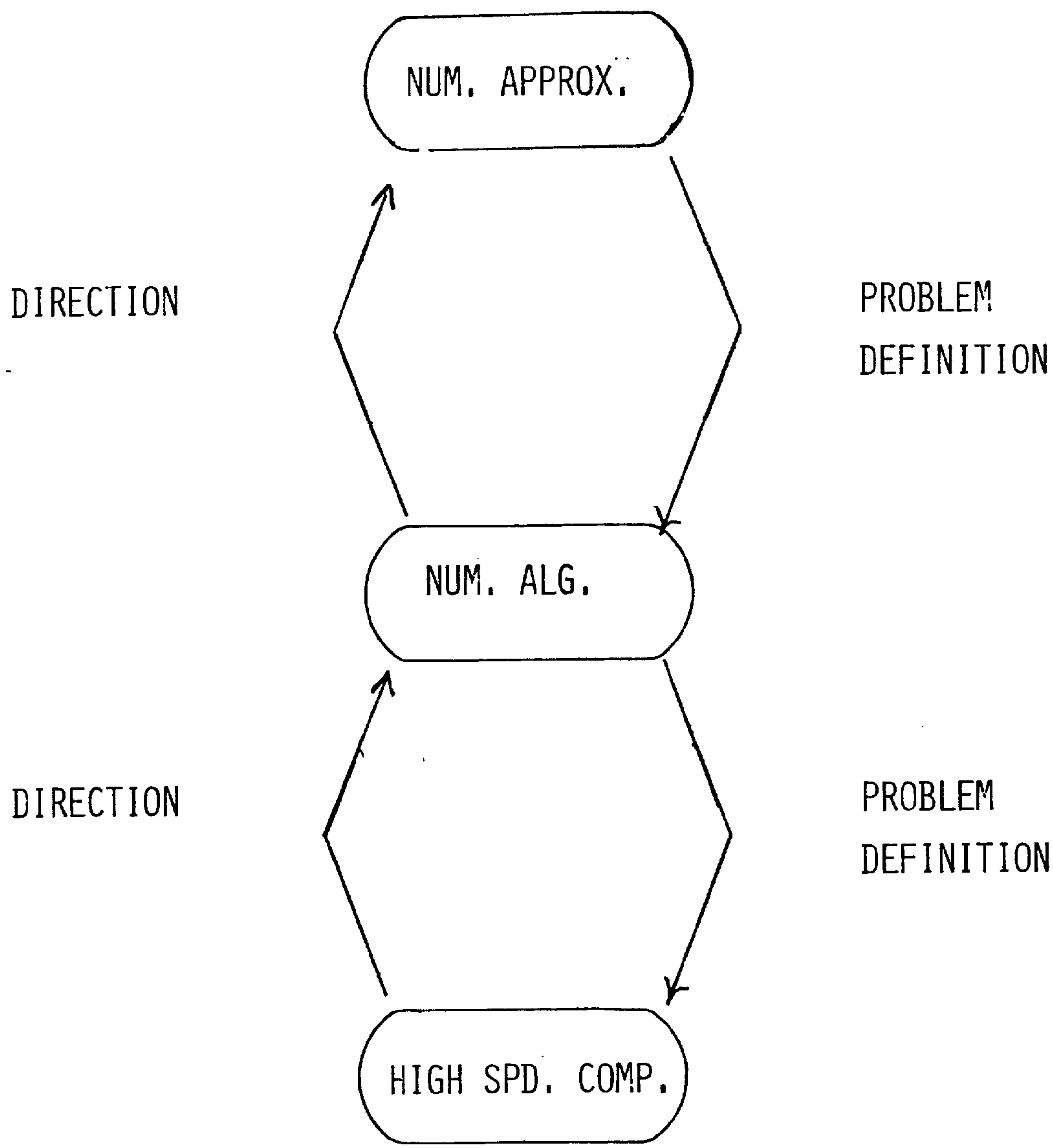
USE VECTOR REGISTERS AS TEMPORARY STORAGE



KEY POINTS TO REMEMBER

- VECTOR COMPUTER HAS GENERATED A NEW MATHEMATICAL TOOL FOR THE NUMERICAL ANALYST
- MORE MATHEMATICAL THEORY NEEDS TO BE DEVELOPED
- COMPILERS CANNOT GENERATE THE OPTIMAL ALGORITHM
- MANY NEW ALGORITHMS HAVE BEEN DEVELOPED

DIFFICULT TO CONVEY INFORMATION TO NEIGHBORING LEVELS

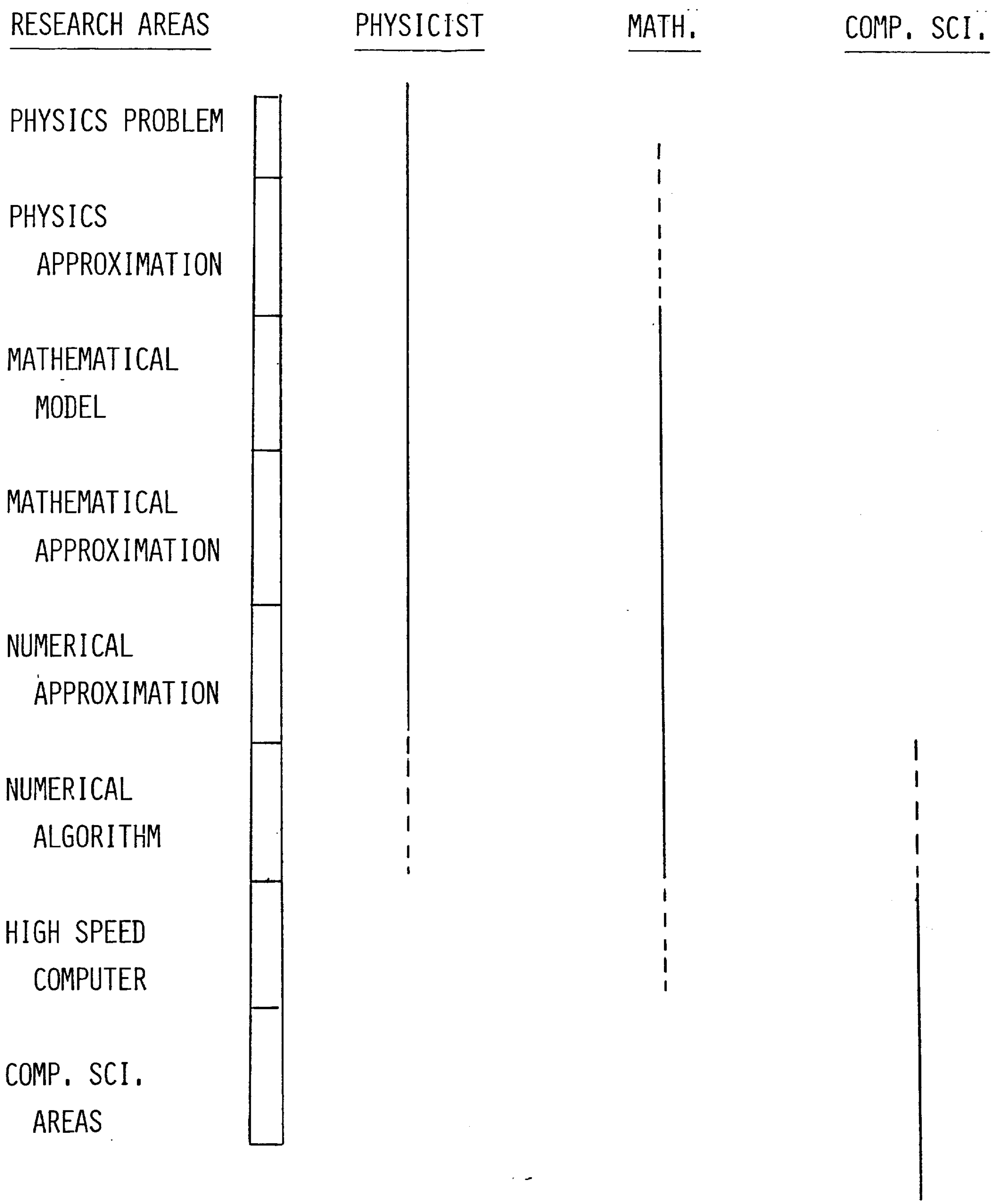


RECALL: THE SCIENTIFIC COMPUTATIONAL PROCESS

PHYSICAL WORLD	PHYSICS PROBLEM
	↓
BRIDGE	PHYSICS APPROXIMATION
	↓
MATHEMATICAL WORLD	MATHEMATICAL MODEL
	↓
	MATHEMATICAL APPROXIMATION
	↓
MATHEMATICAL WORLD	NUMERICAL APPROXIMATION
	↓
BRIDGE	NUMERICAL ALGORITHM
	↓
MACHINE WORLD	CALCULATIONAL DEVICE

MAIN DIFFICULTY IN NUMERICAL ALGORITHM DEVELOPMENT:

COMMUNICATION PROBLEMS



THE BASIC GOAL OF THE EFFORT IN NUMERICAL ALGORITHMS:

MAKE THIS LINK IN COMPUTATIONAL PROCESS AS STRONG AS POSSIBLE

NUMERICAL APPROXIMATION

$\downarrow \uparrow$

NUMERICAL ALGORITHM

$\downarrow \uparrow$

HIGH SPEED
COMPUTER

BY

1) RESEARCH IN NUMERICAL ALGORITHMS
2) PROBLEM DEFINITION TO HIGH SPEED COMPUTER ARCHITECTS
3) FOCUS RESEARCH IN NUMERICAL APPROXIMATIONS

OTHER ACTIVITIES

$$
\begin{tikzcd}
\text{NUMERICAL ALGORITHMS} \arrow[d, "\text{PROBLEM DEFINITION}"] \\
\text{HIGH SPEED COMPUTERS}
\end{tikzcd}
$$

- GENERATING LARGE SCALE COMPUTER CODES FOR RESEARCH PURPOSES BY
  - ACADEMIC
  - INDUSTRY
  - LABORATORY

- THESE CODES WILL CONTAIN ALGORITHMS GENERATED FROM RESEARCH

OTHER ACTIVITIES

NUMERICAL APPROXIMATIONS

$\uparrow$ DIRECTIONS

NUMERICAL
ALGORITHMS

- NEW TIME-DIFFERENCING TECHNIQUES HAVE BEEN DEVELOPED
- NEW RESULTS IN CONJUGATE GRADIENT METHOD

CONCLUDING REMARKS

- COMPUTERS DO INFLUENCE THE TYPE OF NUMERICAL ALGORITHMS THAT ARE USED IN YOUR CALCULATION

- NUMERICAL ALGORITHMS ARE THE BRIDGE BETWEEN THE PHYSICAL WORLD AND THE COMPUTING MACHINE
  - DON'T TRY TO BY-PASS THEM

- TALK TO YOUR FRIENDLY NUMERICAL ANALYST. THEY CAN REALLY HELP YOU TO
  1) BUILD A BETTER COMPUTER
  2) SOLVE YOUR PROBLEM ON AN EXOTIC COMPUTER